0188 S D

DRH 0587-8

ornl

ORNL-6497

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

Modeling Cyclic Melting and Refreezing in a Hollow Metal Canister

D. G. Wilson
R. E. Flanery

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

ORNL--6497

DE89 001325

Engineering Physics and Mathematics Division
Mathematical Sciences Section

MODELING CYCLIC MELTING AND REFREEZING IN A HOLLOW METAL CANISTER

D. G. Wilson and R. E. Flanery

Oak Ridge National Laboratory
P. O. Box 2009
Oak Ridge, Tennessee 37831-8083

Date Published: September 1988

Research sponsored by the
NASA Advanced Solar Dynamics Program
under Interagency Agreement No. 1819-1819-A
NASA order No. C-30001-J
and the
Mathematical Sciences Section
Engineering Physics and Mathematics Division
Oak Ridge National Laboratory

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U. S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400

Table of Contents

Modeling Cyclic Melting and Refreezing in a Hollow Metal Canister

D. G. Wilson, and R. E. Flanery

ABSTRACT

This report documents the mathematical model and computational algorithms used in a pair of computer programs that do energy redistribution calculations as part of a comprehensive simulation for thermal and structural analyses of one component of a thermal energy storage system for the manned space station. The complete problem includes cyclic melting and refreezing, fluid flow, and void formation and movement, as well as conductive and convective heat transfer in a three dimensional setting. The problem is posed in a hollow, metal canister filled with a high temperature phase change material. The heat transfer equations discussed here consist of a pair of partial differential equations for energy transfer (one linear the other mildly nonlinear), coupled with a constitutive relation for energy and temperature. This constitutes a weak, "enthalpy" formulation of the phase change problem. The partial differential equations are approximated by a system of coupled Crank Nicholson-type finite difference equations. These nonlinear, implicit equations are solved for enthalpy (energy content) and temperature fields simultaneously. A successive overrelaxation iteration scheme with red/black ordering is used to solve the nonlinear difference equations. The algorithms have been vectorized for rapid execution on the Cray X-MP supercomputer and techniques used to do this are discussed.

1. INTRODUCTION

It is intended that the manned space station satisfy a considerable portion of its power requirements with solar energy. The station will orbit the earth in about ninety minutes and spend about two thirds of each orbit in sunlight and about one third in the earth's shadow. Under these conditions it will be necessary to store thermal energy during the station's exposure to the sun and to retrieve it during the transit through the earth's shadow. Several systems have been proposed for accomplishing this. In this report we describe in rough outline one such system and in much more detail the mathematical and computational model used in the heat flow portion of a simulation of its performance.

The system considered consists of a solar collector lined with small metal canisters filled with a high temperature phase change material (PCM), lithium fluoride salt. The canisters are small enough to fit comfortably in the palm of one's hand and there are more than a hundred of them. A heat transfer fluid, an inert gas such as helium or neon, circulates through pipes that pass through the metal canisters and carries heat away to supply energy to turbines, generators, etc. The continual melting and refreezing of the PCM distributes over time the delivery of the solar energy to the transfer fluid and hence to the heat engines beyond. The motivation for using a PCM based thermal energy storage system is that a properly sized such system can store and deliver energy over a narrow temperature range near the melting point of the PCM thus avoiding temperature extremes.

A multitude of problems must be solved to design a satisfactory system. To list a few: The size of the solar collector must be matched with the power requirements of the station. The capacity of the energy storage system must be such that the PCM just about completely melts during the insolation period and just about completely freezes during the dark period. (Otherwise the advantage of the PCM is lost.) The canister material and its design must be adequate to withstand the frequent meltings and freezings of the PCM, and the resulting mechanical stresses, for many cycles.

A computer simulation of the continual thermal energy redistribution in a representative canister has been developed for the Advanced Solar Dynamics Program (ASDP) at NASA Lewis Research Center. Several modules, including two containing the model and algorithms described here, make up a code named "NORVEX." This code will be used in support of a series of flight experiments proposed by ASDP to evaluate components of the proposed system.

The complete mathematical problem consists of a system of coupled, mildly nonlinear, partial differential equations for heat flow and fluid flow in a representative canister coupled with a constitutive relation between energy and temperature and equations for development and movement of a vapor filled void. There will be fluid flow in the liquid PCM, even in a microgravity environment, because there is a significant difference in density between solid and liquid PCM. The density change on melting and freezing causes the formation of a vapor filled void on freezing and, of course, the disappearance of the void on remelting. What is described in this document is only that part of the simulation dealing with heat flow in the canister and the enclosed PCM. Except for the little said in this introduction, the description of the fluid flow problem and its numerical solution and the modeling of the void development and movement is left to other reports.

In brief the problem is as follows. A right circular cylindrical canister is filled with a high temperature PCM. An enclosed cylindrical pipe runs down the center of the canister. To simulate the heat transfer fluid flowing in this pipe, and to model the flight experiments proposed by ASDP to evaluate system components, it has been assumed that a solid nickel cylinder is enclosed by and attached to this inner pipe. Periodic flux boundary conditions are supplied at the outer cylindrical surface of the canister and zero flux at its end faces. Internal boundary conditions between canister and PCM, and between canister and enclosed nickel cylinder, insure conservation of energy. (The enclosed nickel cylinder radiates energy to the sky, at a flared end exterior to the canister, via radiative heat

transfer; but descriptions of the simulation of heat transfer in this nickel cylinder and its radiation to the sky are omitted from this report. These, and other aspects of the simulation not described here, are documented in the final report.)

For this document, the fluid velocities, void location and temperature field in the enclosed nickel conductor are assumed to be known at each time step. These are computed in separate modules of the complete simulation. Beyond the following three sentences, these are not discussed in this document. In the fluid flow module, the movement of the liquid PCM is modeled using a weak formulation of the incompressible Navier Stokes equations. The phase transition region is treated as a porous medium that inhibits fluid flow but also introduces the density change that acts as a source for the fluid flow. The boundary conditions on the fluid flow problem are: "no slip" at the walls of the canister, i. e. all velocity components are zero there, and conservation of mass and momentum at the surface of the void.

Heat flow in the containing, metal canister is modeled using the partial differential equation for conductive heat transfer in cylindrical polar coordinates. The boundary conditions on the heat flow problem in the canister are: imposed flux on the outer surfaces, and continuity of flux across the interfaces between canister and PCM and between canister and enclosed nickel cylinder. An "enthalpy formulation" that permits easy treatment of the successive melting and freezing cycles is used to model the heat flow in the PCM. In this formulation, the dependent variable in the partial differential equation is enthalpy content instead of temperature, and a constitutive relation gives temperature as a function of enthalpy. A convection term is included to account for the heat transfer caused by bulk movement of the liquid. The boundary conditions on the heat flow problem in the PCM are just that energy be conserved at the canister walls.

The complete system of coupled, mildly nonlinear, partial differential equations is approximated by a system of coupled Crank Nicholson type finite difference equations.

Implicit equations are solved for the enthalpy (energy content), temperature and velocity fields simultaneously. A successive overrelaxation (SOR) iteration scheme with red/black ordering is used to solve the difference equations in both the canister and the PCM.

In section 2 we present the analytic formulation of the heat transfer problems in the canister and in the PCM. This includes the partial differential equation for heat flow in the canister and the enthalpy formulation in the PCM including the constitutive relation between enthalpy and temperature. Cylindrical polar coordinates are used throughout. In section 3 we present the discrete mesh, the control volumes used and the implicit difference equations that approximate the partial differential equations. This section includes descriptions of the computation of equivalent thermal conductivities between dissimilar materials (canister, liquid PCM, solid PCM, etc.) and the "upwinding" scheme used in the discrete convection terms. This scheme is nonstandard because the mesh used is not uniform in the r and z coordinate directions. In section 4 we discuss the solution of the discrete equations using SOR iterations with red/black ordering. The equations in the canister are linear and the corresponding SOR scheme is standard except for the red/black ordering. The equations in the PCM are nonlinear because the thermal conductivities in solid and liquid differ and because the discrete, nonlinear, constitutive relation between enthalpy and temperature must be satisfied simultaneously with the finite difference equations. Thus the corresponding SOR scheme is nonstandard. The idea for the implicit, discrete form of an enthalpy formulation for a phase change problem is due to Elliot and Ockendon, [1]. However, both its implementation in cylindrical polar coordinates and its application to a problem involving convection are new. In section 5 we discuss strategies used to vectorize the algorithms for fast execution on the Cray X-MP supercomputer. Logical variables were used as multipliers to avoid conditional branching in loops that would prevent vectorization; multidimensional arrays were "unfolded" to increase the average length of vectors; red/black ordering of nodes was used to avoid vector

dependencies; and "phantom" node points were inserted into arrays to handle boundary conditions in the midst of these long, unfolded vectors. These are applications and extensions of ideas used in the numerical solution of Stefan type problems in a three dimensional, rectangular parallelepiped [3]. A paper documenting these methods, for application to more general finite difference schemes, is in preparation [2].

2. ANALYTIC FORMULATION OF THE HEAT TRANSFER PROBLEMS

A. Heat Flow in the Canister.

The partial differential equation for heat conduction used to model the heat flow in the containing canister is

$$\rho c \frac{\partial T}{\partial t} = div\,(k\,\nabla T), \tag{1}$$

where ρ, c, and k are the density, heat capacity and thermal conductivity of the canister material respectively, and T is the temperature. In cylindrical polar coordinates, this is

$$\rho c \frac{\partial T}{\partial t} = \frac{1}{r}\frac{\partial}{\partial r}(r\,k\,\frac{\partial T}{\partial r}) + \frac{1}{r^2}\frac{\partial}{\partial \theta}(k\,\frac{\partial T}{\partial \theta}) + \frac{\partial}{\partial z}(k\,\frac{\partial T}{\partial z}), \tag{2}$$

for r, θ, and z inside the canister. The canister is identified as the union of the four sets LE (left end), RE (right end), IC (inside cylinder) and OC (outside cylinder) defined by

$$LE = \{r, \theta, z \mid R_{inner} < r < R_{outer}, 0 < \theta < 2\pi, Z_{left} < z < Z_1\},$$

$$RE = \{r, \theta, z \mid R_{inner} < r < R_{outer}, 0 < \theta < 2\pi, Z_{right-inner} < z < Z_{right}\},$$

$$IC = \{r, \theta, z \mid R_{inner} < R < R_1, 0 < \theta < 2\pi, Z_{left} < z < Z_{right}\},$$

$$OC = \{r, \theta, z \mid R_{inner-outer} < r < R_{outer}, 0 < \theta < 2\pi, Z_{left} < z < Z_{right}\},$$

where R_{inner} and R_1 denote the radii of the inner and outer walls of the inside cylinder of the canister respectively, $R_{inner-outer}$ and R_{outer} denote the radii of the inner and outer walls of the outside cylinder of the canister respectively, Z_{left} and Z_1 denote the axial coordinates of the inner and outer walls of the outside disk of the canister at the left

respectively, and $Z_{right-inner}$ and Z_{right} denote the axial coordinates of the inner and outer walls of the outside disk of the canister at the right respectively.

Flux boundary conditions are imposed at three of the four exterior boundaries of the canister. The end walls are insulated and the incoming energy flux on the outer cylindrical wall is specified. ($\frac{\partial T^+}{\partial z} = 0$ at $z = Z_{left}$, $\frac{\partial T^-}{\partial z} = 0$ at $z = Z_{right}$, and $-k_{canister} \frac{\partial T^-}{\partial r} = q(t)$ at $r = R_{outer}$.) At the fourth exterior boundary, the inner surface of the inner cylindrical wall, it is only required that the energy flux be continuous across the interface between the canister and the enclosed nickel cylinder. ($-k_{canister} \frac{\partial T^+}{\partial r} = -k_{nickel} \frac{\partial T^-}{\partial r}$ at $r = R_{inner}$.) At the interior boundaries of the canister, where it contacts the PCM, it is only required that the energy flux be continuous across the interface between the canister and the enclosed PCM. $-k_{canister} \frac{\partial T^-}{\partial r} = -k_{PCM} \frac{\partial T^+}{\partial r}$ at $r = R_1$, $-k_{canister} \frac{\partial T^-}{\partial r} = -k_{PCM} \frac{\partial T^+}{\partial r}$ at $r = R_{inner-outer}$, and $-k_{canister} \frac{\partial T^-}{\partial z} = -k_{PCM} \frac{\partial T^+}{\partial z}$ at $z = Z_1$, and $-k_{canister} \frac{\partial T^+}{\partial z} = -k_{PCM} \frac{\partial T^-}{\partial z}$ at $z = Z_{right-inner}$.) Here $k_{canister}$, k_{nickel}, and k_{PCM} denote the thermal conductivities of canister, nickel, and PCM respectively, and the superscripts on the partial derivatives of the temperature indicate limiting values from above and below.

B. Energy Redistribution in the PCM.

An enthalpy formulation that permits easy treatment of the successive melting and freezing cycles is used to model the heat flow in the PCM. A convection term in the equations accounts for the heat transfer caused by movement of the liquid PCM. The partial differential equation used to model the heat flow in the PCM is

$$\frac{\partial(\rho e)}{\partial t} = div(k \nabla T) - div(\rho e \mathbf{v}), \qquad (3)$$

where ρ is the density of the PCM, which is substantially different in the solid and liquid phases, e is the specific enthalpy, k is the thermal conductivity of the PCM (also different in solid and liquid phases), T is the temperature, and v is the velocity. The enthalpy formulation consists of equation (3) coupled with the following constitutive relation between ρe and T.

$$T = \begin{cases} T_{melt} + \dfrac{(\rho e - \rho_{liquid} H)}{\rho_{liquid} c_{liquid}} & \text{for } \rho_{liquid} H < \rho e , \\ T_{melt} & \text{for } 0 \leqslant \rho e \leqslant \rho_{liquid} H , \\ T_{melt} + \dfrac{\rho e}{\rho_{solid} c_{solid}} & \text{for } \rho e < 0 , \end{cases} \tag{4}$$

where T_{melt} is the melt temperature of the PCM, ρ_{liquid} and ρ_{solid} are the densities of the liquid and solid PCM respectively, c_{liquid} and c_{solid} are the heat capacities of the liquid and solid PCM respectively, and H is the latent heat of the solid liquid phase transition.

Interactions with the containing canister define the boundary conditions for both the heat flow and fluid flow problems in the region occupied by the PCM. At these boundaries energy must be conserved, and so the energy flux must be continuous across the interface between the canister and the PCM. The equations embodying this condition are given at the end of the previous subsection. The velocity boundary condition is that the liquid PCM does not slip at the boundary with the canister, that is, all components of the velocity are zero at the inner surfaces of the canister walls.

3. FORMULATION OF DISCRETE EQUATIONS

A. Definitions of the discrete mesh and control volumes.

A finite mesh of r, θ and z values is defined on the canister. The coordinate values are r_i, for $i = 0, 1, \cdots, I$, θ_j, for $j = 0, 1, \cdots, J$, and z_k, for $k = 0, 1, \cdots, K$. For parity reasons having to do with the red/black ordering, I, J, and K, the number of subdivisions in the r, θ and z coordinate directions respectively, are each required to be odd. With R_{inner}, R_{outer}, Z_{left}, and Z_{right} as before, we have $r_0 = R_{inner}$, $r_I = R_{outer}$, $z_0 = Z_{left}$, and $z_K = Z_{right}$. Similarly $\theta_0 =$ zero and $\theta_J = 2\pi$. We think of the canister, and also the enclosed region filled with PCM, as being divided into small "control volumes" whose edges are defined by the mesh lines. The mesh is not required to be uniform in the r and z directions, and the mesh we use is not uniform in these directions. We have assumed the mesh to be uniform in the θ direction. Thus θ takes $J+1$ equally spaced values from zero to 2π inclusive and $\Delta\theta = 2\pi/J$.

At the center of each control volume we identify a "node," and we index these nodes by the indices i, j, and k for $i = 0, 1, \cdots, I-1$, $j = 0, 1, \cdots, J-1$, and $k = 0, 1, \cdots, K-1$. The coordinates of the node p_{ijk} are $(r_{i+1} + r_i)/2$, $(\theta_{j+1} + \theta_j)/2$, $(z_{k+1} + z_k)/2$. The distances in each coordinate direction between an interior node, p_{ijk}, and its next nearest neighbors, $p_{(i+1)jk}$, $p_{i(j+1)k}$, and $p_{ij(k+1)}$, are $(r_{i+2} - r_i)/2$, $(\theta_{j+2} - \theta_j)(r_{i+1} + r_i)/4$, (or, since the angular mesh is uniform, $\Delta\theta\,(r_{i+1} + r_i)/2$), and $(z_{k+2} - z_k)/2$ respectively.

We think of discrete temperature, density and enthalpy variables, T_{ijk}, ρ_{ijk} and e_{ijk}, as being associated with the control volumes. We think of discrete velocity variables in each coordinate direction, vr_{ijk}, $v\theta_{ijk}$ and vz_{ijk} as being associated with the underlying partition that defines the interfaces between control volumes. Thus vr^n_{ijk} is the discrete approximation to the radial fluid flow velocity at time level n at $r = r_i$, $\theta = (\theta_{j+1} + \theta_j)/2$, $z = (z_{k+1} + z_k)/2$, (this is the center of the interface between cells

numbered $(i\text{-}1)jk$ and ijk); $v\theta_{ijk}$ is the discrete approximation to the azimuthal fluid flow velocity at time level n at $\theta = \theta_j$, $r = (r_{i+1} + r_i)/2$, $z = (z_{k+1} + z_k)/2$, (this is the center of the interface between cells numbered $i(j\text{-}1)k$ and ijk); and vz_{ijk} is the discrete approximation to the axial fluid flow velocity at time level n at $z = z_k$, $\theta = (\theta_{j+1} + \theta_j)/2$, $r = (r_{i+1} + r_i)/2$, (this is the center of the interface between cells numbered $ij(k\text{-}1)$ and ijk). Figure 1 shows a typical control volume and the locations of the six velocity components into and out of this cell.

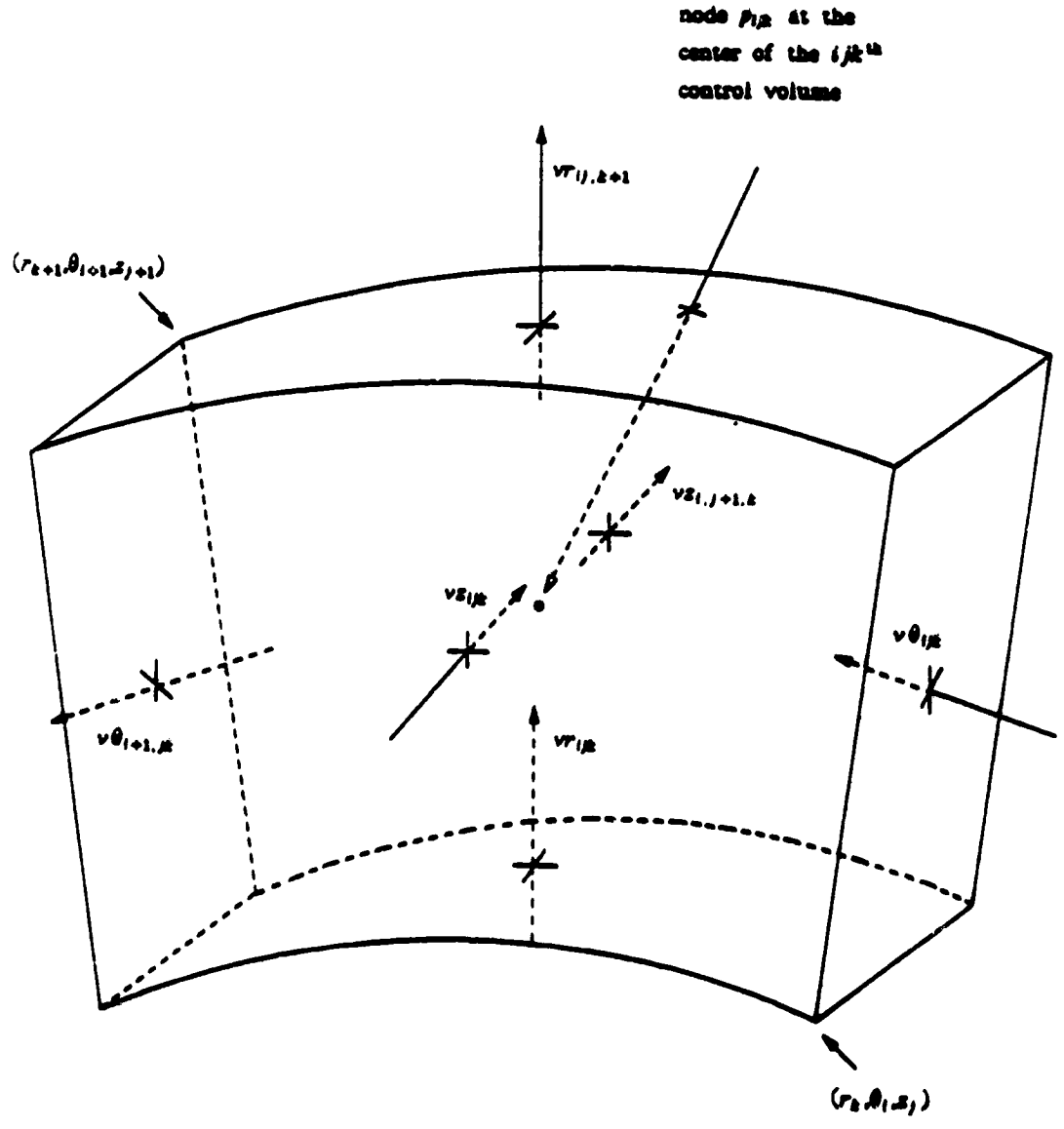

Figure 1. A typical control volume.

B. Discrete Equations for Heat Flow in the Canister.

For simulating heat flow in the canister, we have assumed that the walls of the canister are only one control volume thick. The equations used are a Crank Nicholson like, finite difference discretization of the partial differential equation (2). The equations are of the form

$$\frac{T^{n+1}_{ijk} - T^{n}_{ijk}}{\Delta t} = \Theta \left| \text{terms at } (n+1) \right| + (1-\Theta) \left| \text{terms at } n \right|. \qquad (5)$$

for ijk running through all the control volumes in the canister. Since we have assumed that the walls of the canister are only one cell thick, these are: $i = 0$ and $i = I-1$ with $j = 0, 1, \cdots, J-1$ and $k = 0, 1, \cdots, K-1$, for the inner and outer cylindrical walls respectively; and $k = 0$ and $k = K-1$ with $i = 1, 2, \cdots, I-2$ and $j = 0, 1, \cdots, J-1$, for the end walls respectively. The parameter Θ is a number between zero and one that gives the implicitness of the difference scheme. Θ is an input parameter, but we have uniformly taken it to be one half, resulting in a Crank Nicholson scheme. The terms in the parentheses in equation (5) are difference quotients in the three coordinate directions corresponding to the terms on the right of equation (2).

The difference quotients in the r direction at time level n are

$$(\rho c)^{-1} \left| r_{i+1} \mathrm{k} \frac{T^{n}_{(i+1)jk} - T^{n}_{ijk}}{(r_{i+2} - r_i)} - r_i \mathrm{k} \frac{T^{n}_{ijk} - T^{n}_{(i-1)jk}}{(r_{i+1} - r_{i-1})} \right| / \left| (r_{i+1} + r_i)(r_{i+1} - r_i) \right| \qquad (6)$$

where ρ, c, and k are the density, heat capacity and thermal conductivity of the canister material respectively. The difference quotients in the θ direction at time level n are

$$(\rho c)^{-1} \left| \mathrm{k} \frac{T^{n}_{i(j+1)k} - T^{n}_{ijk}}{(\theta_{j+1} - \theta_j)} - \mathrm{k} \frac{T^{n}_{ijk} - T^{n}_{i(j-1)k}}{(\theta_j - \theta_{j-1})} \right| / \left| (\theta_{j+1} - \theta_j) \left| (r_{i+1} + r_i)/2 \right|^2 \right| \qquad (7)$$

Since we have assumed that the grid is uniform in the θ direction, all the θ differences are equal ($\Delta\theta = 2\pi/J$). The difference quotients in the z direction at time level n are

$$(\rho c)^{-1} \left| \mathrm{k} \frac{T^{n}_{ij(k+1)} - T^{n}_{ijk}}{(z_{k+2} - z_k)/2} - \mathrm{k} \frac{T^{n}_{ijk} - T^{n}_{ij(k-1)}}{(z_{k+1} - z_{k-1})/2} \right| / (z_{k+1} - z_k). \qquad (8)$$

No single difference equation for a cell in the canister has all three of these terms. This is because the walls of the canister are considered to be only one cell thick and thus every cell is affected by one or more boundary conditions. Each cell in the canister interacts with canister material at four of its six interfaces, but at the other two interfaces one of the boundary conditions holds. Cells in the ends of the canister have an insulated barrier at the exterior face, cells in the outer cylinder have a specified incoming flux at the exterior face, and cells in the inner cylinder have the nickel cylinder at the exterior face. All cells in the canister, except those in the inner and outer rings of cells at the ends, have PCM at an interior face. The two inner and outer rings at the ends of the canister (four in all) have canister material at both interior interfaces and do not interact with the PCM.

Since cells in the left and right end walls have insulating boundaries to the left and right respectively, the terms corresponding to $\left\{ k \frac{T^n_{ij\,0} - T^n_{ij\,(-1)}}{(z_0 - z_{-1})/2} \right\}$ and $\left\{ k \frac{T^n_{ij\,K} - T^n_{ij\,(K-1)}}{(z_{K+1} - z_{K-1})/2} \right\}$ are replaced by zero. For these same cells, except the innermost and outermost rings that do not interact with the PCM, the terms corresponding to $\left\{ k \frac{T^n_{ij\,1} - T^n_{ij\,0}}{(z_2 - z_0)/2} \right\}$ and $\left\{ k \frac{T^n_{ij\,(K-1)} - T^n_{ij\,(K-2)}}{(z_K - z_{K-2})/2} \right\}$, where the canister contacts the PCM, are replaced by $\left\{ k_{ij\,(0+\frac{1}{2})} \frac{T^n_{ij\,1} - T^n_{ij\,0}}{(z_2 - z_0)/2} \right\}$ and $\left\{ k_{ij\,(K-1-\frac{1}{2})} \frac{T^n_{ij\,(K-1)} - T^n_{ij\,(K-2)}}{(z_K - z_{K-2})/2} \right\}$ respectively where now $k_{ij\,(0+\frac{1}{2})}$ and $k_{ij\,(K-1-\frac{1}{2})}$ are equivalent thermal conductivities between PCM and cannister, and $T^n_{ij\,1}$ and $T^n_{ij\,(K-2)}$ are temperatures in the PCM. We defer for a moment explaining the computation of the equivalent thermal conductivity between PCM and canister.

For cells in the exterior cylindrical walls, the terms corresponding to $\left\{ k \frac{T^n_{Ijk} - T^n_{(I-1)jk}}{(r_{I+1} - r_{I-1})/2} \right\}$ are replaced by the negative of the incoming flux and, for all except the first and last rings of cells that do not interact with the PCM, the terms corresponding

to $\left\{ k \frac{T^n_{(I-1)jk} - T^n_{(I-2)jk}}{(r_{I+1} - r_{I-1})/2} \right\}$, where the canister contacts the PCM, are replaced by $\left\{ k_{(I-1-\frac{1}{2})jk} \frac{T^n_{(I-1)jk} - T^n_{(I-2)jk}}{(r_I - r_{I-2})/2} \right\}$. Here, as before, $k_{(I-1-\frac{1}{2})jk}$ is an equivalent thermal conductivity between PCM and cannister, and $T^n_{(I-2)jk}$ is a temperature in the PCM.

For cells in the interior cylindrical walls, the terms corresponding to $\left\{ k \frac{T^n_{1jk} - T^n_{0jk}}{(r_2 - r_0)/2} \right\}$, where the cannister contacts the PCM (which again excludes the first and last rings of cells), are replaced by $\left\{ k_{(0+\frac{1}{2})jk} \frac{T^n_{1jk} - T^n_{0jk}}{(r_2 - r_0)/2} \right\}$, where $k_{(0+\frac{1}{2})jk}$ is an equivalent thermal conductivity between the PCM and the cannister, and T^n_{1jk} is a temperature in the PCM, and the terms corresponding to $\left\{ k \frac{T^n_{0jk} - T^n_{(-1)jk}}{(r_1 - r_{-1})/2} \right\}$, are replaced by $\left\{ k_{(0-\frac{1}{2})jk} \frac{T^n_{0jk} - T^n_{(-1)jk}}{(r_1 - r_{-1})/2} \right\}$, where $k_{(0-\frac{1}{2})jk}$ is an equivalent thermal conductivity between cannister and the nickel core inside the central cylinder, $T^n_{(-1)jk}$ is a temperature in the outermost ring of cells of the nickel core and r_{-1} is the radial coordinate of the inside edge of this ring of cells in the core.

In the angular direction we have a periodic boundary condition. That is, the interface at $\theta = 0$ is identified with the the interface at $\theta = 2\pi$. Thus, in the θ direction, the neighbors of the first cell in the j direction ($j = 0$) are the second cell and the last cell and, similarly, the neighbors of the last cell in the j direction ($j = J-1$) are the next to last cell and the first cell. Thus, in the equations for cells with $j = 0$, the entries in equations (5) and (7) with subscripts j-1 are replaced with entries with subscripts J-1, and similarly, in the equations for cells with $j = J-1$, the entries in equations (5) and (7) with subscripts $j+1$ are replaced with entries with subscripts $j = 0$.

Equivalent thermal conductivities between dissimilar materials are computed using an equivalent thermal resistance model. An expression for the equivalent thermal

conductivities can be derived by considering the thermal energy flow between two juxtaposed slabs of dissimilar materials with different thicknesses. The condition to be satisfied is that the flux of energy is continuous across the boundary between the slabs. The equations are as follows.

$$-k_{eq}\frac{T_2 - T_1}{(w_2 + w_1)/2} = -k_1\frac{T_{inter} - T_1}{(w_1/2)} = -k_2\frac{T_2 - T_{inter}}{(w_2/2)},$$

where T_1, T_2 and T_{inter} are the temperatures associated with the two slabs and with the interface where they touch respectively, w_1 and w_2 are the widths of the slabs, k_1 and k_2 are the thermal conductivities of the materials that make up the slabs, and k_{eq} is the equivalent thermal conductivity between them. This gives two independent linear equations in the two unknowns T_{inter} and k_{eq}. Since the quantity T_{inter} is of no particular interest, we do not solve for it. The expression for the desired equivalent thermal conductivity is

$$k_{eq} = \frac{k_1 k_2 (w_1 + w_2)}{w_2 k_1 + w_1 k_2}. \tag{9}$$

This is the equation used to compute thermal conductivities between canister and PCM, between canister and the nickel core, and between different phases of PCM.

C. Discrete Equations for Energy Redistribution in the PCM.

The discretization of the enthalpy equations, (3) and (4), is similar to that for equation (2). However, since ρ, the material density and e, the specific enthalpy, vary together (it is their product that occurs in the governing partial differential equations), we take this product to be a single variable, denoted by "ρe." In the following discussion, "ρe" is to be interpreted as a multicharacter symbol for this variable, and ρe^n_{ijk} is an approximation to the integral average of its value over the ijkth cell at time n. This is an approximation to the product ρe, but densities and enthalpies are not computed separately.

As before, the discretization of the partial differential equation, (3), takes the form

$$\frac{\rho e_{ijk}^{n+1} - \rho e_{ijk}^{n}}{\Delta t} = \Theta \left[\text{ terms at } (n+1) \right] + (1-\Theta) \left[\text{ terms at } n \right], \qquad (10)$$

for ijk running through all the control volumes in the PCM, $i = 1, 2, \cdots, I-2$, $j = 0, 1, \cdots, J-1$, and $k = 1, 2, \cdots, K-2$, (which includes those filled only with vapor, i. e. void, and those located next to canister walls). The parameter Θ in equation (10) is the same parameter that appears in equation (5), a number between zero and one that gives the implicitness of the difference scheme. We have uniformly taken Θ to be one half, which gives a Crank Nicholson scheme. In addition to this discrete version of the partial differential equation, the following discrete version of the constitutive relation must be satisfied.

$$T_{ijk}^{n+1} = \begin{cases} T_{melt} + \dfrac{(\rho e_{ijk}^{n+1} - \rho_{liquid} H)}{\rho_{liquid} c_{liquid}} & \text{for } \rho_{liquid} H < \rho e_{ijk}^{n+1}, \\ T_{melt} & \text{for } 0 \leqslant \rho e_{ijk}^{n+1} \leqslant \rho_{liquid} H, \\ T_{melt} + \dfrac{\rho e_{ijk}^{n+1}}{\rho_{solid} c_{solid}} & \text{for } \rho e_{ijk}^{n+1} < 0. \end{cases} \qquad (11)$$

The terms that appear in the parentheses in equation (10) are differences of incoming and outgoing fluxes in the three coordinate directions for heat conduction and fluid flow. These involve values of T, ρe and v. Because the PCM is completely surrounded by the canister, the externally imposed boundary conditions do not affect the equations in the PCM. The form of the equations for cells in the PCM that are neighbors of cells in the canister is the same as that for interior cells. The only differences are that the thermal conductivity of the canister material is used in equation (9) to compute the appropriate equivalent thermal conductivities between adjacent cells, and the temperatures of cells in the canister are known. Because in the fluid flow terms we use "upwinding" and this requires a separate discussion, we present first only the conductive heat transfer terms.

The terms for conductive heat transfer in the r direction at time level n are

$$\frac{r_{i+1}\,k^{n}_{(i+\frac{1}{2})jk}\,\dfrac{T^{n}_{(i+1)jk}-T^{n}_{ijk}}{(r_{i+2}-r_i)} \;-\; r_i\,k^{n}_{(i-\frac{1}{2})jk}\,\dfrac{T^{n}_{ijk}-T^{n}_{(i-1)jk}}{(r_{i+1}-r_{i-1})}}{(r_{i+1}+r_i)(r_{i+1}-r_i)}. \tag{12}$$

where $k^{n}_{(i+\frac{1}{2})jk}$ and $k^{n}_{(i-\frac{1}{2})jk}$ are equivalent thermal conductivities between the cell numbered ijk and its neighbors in the radial direction numbered $(i+1)jk$ and $(i-1)jk$ respectively. These are computed using equation (9) where the dissimilar materials are the contents of the adjacent cells at time level n (and thus the superscript) which may be any two from among: canister material, liquid PCM, solid PCM, "mush" (partly liquid and partly solid PCM), and void (whose thermal conductivity is taken to be zero).

A cell is mushy by definition if, for that cell, $0 < \rho e < \rho_{liquid} H$, where ρ_{liquid} is the density of the liquid PCM, and H is the latent heat of the liquid/solid phase transition of the PCM. The thermal conductivity of a mushy cell is computed as a linear combination of the thermal conductivities of liquid and solid PCM in which the coefficients are the fraction of liquid and solid present respectively. The liquid fraction is given by $\lambda = \rho e/(\rho_{liquid} H)$ when $0 < \rho e < \rho_{liquid} H$. ($\lambda$ is zero when $\rho e \leq 0$ and one when $\rho e \geq \rho_{liquid} H$.)

The terms for conductive heat transfer in the θ direction at time level n are

$$\frac{k^{n}_{i(j+\frac{1}{2})k}\,\dfrac{T^{n}_{i(j+1)k}-T^{n}_{ijk}}{(\theta_{j+1}-\theta_j)} \;-\; k^{n}_{i(j-\frac{1}{2})k}\,\dfrac{T^{n}_{ijk}-T^{n}_{i(j-1)k}}{(\theta_j-\theta_{j-1})}}{(\theta_{j+1}-\theta_j)\left[(r_{i+1}+r_i)/2\right]^2}. \tag{13}$$

where $k^{n}_{i(j+\frac{1}{2})k}$ and $k^{n}_{i(j-\frac{1}{2})k}$ are equivalent thermal conductivities between the cell numbered ijk and its neighbors in the azimuthal direction numbered $i(j+1)k$ and $i(j-1)k$ respectively. These equivalent thermal conductivities are computed using equation (9) as in the previous discussion, except that, because of the periodic boundary condition in the azimuthal direction, here "canister" is not a possibility for one of the dissimilar materials. Again, since we have assumed that the grid is uniform in the θ direction, all the θ

differences are equal to $2\pi/J$.

The terms for conductive heat transfer in the z direction at time level n are

$$\left\{ k^n_{ij(k+\frac{1}{2})} \frac{T^n_{ij(k+1)} - T^n_{ijk}}{(z_{k+2} - z_k)/2} - k^n_{ij(k-\frac{1}{2})} \frac{T^n_{ijk} - T^n_{ij(k-1)}}{(z_{k+1} - z_{k-1})/2} \right\} / (z_{k+1} - z_k). \quad (14)$$

where $k^n_{ij(k+\frac{1}{2})}$ and $k^n_{ij(k-\frac{1}{2})}$ are equivalent thermal conductivities between the cell numbered ijk and its neighbors in the axial direction numbered $ij(k+1)$ and $ij(k-1)$ respectively. These equivalent thermal conductivities are computed using equation (9) as explained in the discussion of the r difference quotients following equation (12).

The flux terms for fluid flow at the faces of the ijkth cell in the r, θ and z directions at time level n are of the form

$$\frac{r_{i+1}\, \rho e^n_{(i+\frac{1}{2})jk}\, vr^n_{(i+1)jk} - r_i\, \rho e^n_{(i-\frac{1}{2})jk}\, vr^n_{ijk}}{(r_{i+1} - r_i)(r_{i+1} + r_i)/2}. \quad (15a)$$

$$\frac{\rho e^n_{(j+\frac{1}{2})k}\, v\theta^n_{(j+1)k} - \rho e^n_{(j-\frac{1}{2})k}\, v\theta^n_{ijk}}{\Delta\theta\, (r_{i+1} + r_i)/2}, \quad (15b)$$

and

$$\frac{\rho e^n_{ij(k+\frac{1}{2})}\, vz^n_{ij(k+1)} - \rho e^n_{ij(k-\frac{1}{2})}\, vz^n_{ijk}}{z_{k+1} - z_k}, \quad (15c)$$

respectively. Here vr, $v\theta$, and vz are the fluid velocities in the r, θ and z directions respectively, $\Delta\theta$ is the constant angular increment, $2\pi/J$, and the quantities with the half integral subscripts are interpolated values of ρe at the interfaces between adjacent cells. The "no slip" boundary condition at the canister walls implies that all components of the velocities are zero at the smallest and largest values of r and z. However, the discrete velocities are associated with locations in the centers of the faces of cells in the PCM. (See figure 1.) The discrete "no slip" boundary conditions are that all vr's are zero at the largest and smallest values of i and all vz's are zero at the largest and smallest values of k. (None of the $v\theta$'s are *required* to be zero.)

Two things contribute to the computation of the interpolated ρe quantities. These are

the nonuniformity of the grid and possible "upwinding." In the absence of upwinding, we would take the linear interpolant, $\rho e_{m+\frac{1}{2}} = \beta \rho e_{m+1} + (1-\beta)\rho e_m$, where β is the ratio of the distance from the cell interface to the mth node to the total distance from the mth node to the $(m+1)$th node. Here, and for the next few paragraphs, we suppress the time dependence and any distinction among coordinate directions and in place of multiple indices for subscripts use only "m" to denote a single, generic, varying index.

Upwinding is a tactic of weighting the "upwind" quantity more heavily in the computation of the coefficient of the velocity. Numerically this introduces dissipation that tends to reduce the waves created by the hyperbolic character of this term in the differential equation. A physically motivated justification is that the coefficient is partially carried along by the velocity. The usual upwinding formula, for a uniform grid, is

$$\rho e_{m+\frac{1}{2}} v_{m+1} = \rho e_{m+1}(v_{m+1} - \mu \mid v_{m+1} \mid) + \rho e_m (v_{m+1} + \mu \mid v_{m+1} \mid),$$

where $\mu \in [0, 1]$ is the degree of upwinding. With $\mu = 1$ we have full upwinding and $\rho e_{m+\frac{1}{2}}$ is either ρe_m or ρe_{m+1} depending on the sign of v_{m+1}.

A simple alternate formula combining upwinding and unequal cell sizes, made up by taking a weighted linear combination of the terms on the right of the previous formula, would be

$$\rho e_{m+\frac{1}{2}} v_{m+1} = \beta \rho e_{m+1}(v_{m+1} - \mu \mid v_{m+1} \mid) + (1-\beta)\rho e_m (v_{m+1} + \mu \mid v_{m+1} \mid).$$

But this has serious shortcomings. In particular, if β is not one half, it gives obviously incorrect answers both when $\rho e_{m+1} = \rho e_m$,

$$\rho e_{m+\frac{1}{2}} v_{m+1} = \rho e_{m+1} \left\{ v_{m+1} + \mu (1 - 2\beta) \mid v_{m+1} \mid \right\},$$

and when $\mu = 1$,

$$\rho e_{m+\frac{1}{2}}\, v_{m+1} = \begin{cases} 2\beta\, \rho e\, v_{m+1}, & \text{if } v_{m+1} > 0, \\ 2(1-\beta)\, \rho e_{m+1} v_{m+1}, & \text{if } v_{m+1} < 0. \end{cases}$$

The formula we use is

$$\rho e_{m+\frac{1}{2}}\, v_{m+1} = \rho e_{m+1} \left\{ \beta(1-\mu) v_{m+1} + \mu(v_{m+1} - |v_{m+1}|)/2 \right\} \qquad (16)$$
$$+ \rho e_m \left\{ (1-\beta)(1-\mu) v_{m+1} + \mu(v_{m+1} + |v_{m+1}|)/2 \right\}.$$

where μ is the degree of upwinding and β is the ratio of the width of the mth cell to the sum of the widths of the mth and $(m+1)$th cells. (Since nodes are located at centers of cells, this defines the same β as before.)

Writing this explicitly for $v_{m+1} > 0$ and $v_{m+1} < 0$, and rearranging slightly, gives

$$\rho e_{m+\frac{1}{2}}\, v_{m+1} = \begin{cases} \left\{ \beta(1-\mu) \rho e_{m+1} + (1 - \beta(1-\mu)) \rho e_m \right\} v_{m+1}, & \text{if } v_{m+1} > 0, \\ \left\{ \left(1 - (1-\beta)(1-\mu)\right) \rho e_{m+1} + (1-\beta)(1-\mu) \rho e_m \right\} v_{m+1}, & \text{if } v_{m+1} < 0, \end{cases}$$

which shows that for μ and β in (0, 1) the coefficient of the velocity is always a convex linear combination of the ρe values at the neighboring nodes. Thus $\rho e_{m+\frac{1}{2}}$ is always between ρe_m and ρe_{m+1}, and when these agree, $\rho e_{m+\frac{1}{2}}$ equals their common value. When β is one half, equation (16) gives the usual upwinding formula. When μ is zero, equation (16) gives the linear interpolant for the coefficient of the velocity. When μ is one, equation (16) gives full upwinding, and for any positive μ, upwinding persists even in the limits as $\beta \to 0$ and $\beta \to 1$.

There is a considerable lore, but little theory, associated with a strategy for selecting a "good" value for μ. Even this lore is only applicable in one dimension. A general

guideline is, "$\mu > 0$ is necessary but μ as small as possible is good." Obviously μ should depend on the velocity, but the form of this dependence is unknown. We made the value of μ an input parameter. In the absence of a reasonable selection criterion, we took its value to be one half.

To succinctly state the several flux terms corresponding to fluid flow in equation (10), we define a function whose evaluation formula is the right hand side of equation (16).

$$F(\mu,\beta,\phi,\psi,\nu) = \phi\left\{\beta(1-\mu)\nu + \mu(\nu - |\nu|)/2\right\} + \psi\left\{(1-\beta)(1-\mu)\nu + \mu(\nu + |\nu|)/2\right\}. \quad (17)$$

Our intention is to substitute the degree of upwinding for μ, and to substitute ρe and velocity values for ϕ, ψ and ν respectively. To supply appropriate β values for this function, we define parameters, $\beta r_i = (r_i - r_{i-1})/(r_{i+1} - r_{i-1})$, and $\beta z_k = (z_k - z_{k-1})/(z_{k+1} - z_{k-1})$. βr and βz are ratios of distances in the r and z directions respectively. We define $\beta\theta$ to be one half since we have assumed that the mesh is uniform in the angular direction.

With these definitions, the terms of equations (15a) - (15c) are as follows. The radial terms are

$$\frac{r_{i+1}F(\mu,\beta r_i,\rho e^{n}_{(i+1)jk},\rho e^{n}_{ijk},vr^{n}_{(i+1)jk}) - r_i F(\mu,\beta r_{i-1},\rho e^{n}_{ijk},\rho e^{n}_{(i-1)jk},vr^{n}_{ijk})}{(r_{i+1}-r_i)(r_{i+1}+r_i)/2}, \quad (18a)$$

the angular terms are

$$\frac{F(\mu,\beta\theta,\rho e^{n}_{(j+1)k},\rho e^{n}_{ijk},v\theta^{n}_{(j+1)k}) - F(\mu,\beta\theta,\rho e^{n}_{ijk},\rho e^{n}_{(j-1)k},v\theta^{n}_{ijk})}{\Delta\theta\,(r_{i+1}+r_i)/2}, \quad (18b)$$

and the axial terms are

$$\frac{F(\mu,\beta z_k,\rho e^{n}_{ij(k+1)},\rho e^{n}_{ijk},vz^{n}_{ij(k+1)}) - F(\mu,\beta z_{k-1},\rho e^{n}_{ijk},\rho e^{n}_{ij(k-1)},vz^{n}_{ijk})}{(z_{k+1}-z_k)}. \quad (18c)$$

4. SOLUTION OF THE DISCRETE EQUATIONS

A. Discrete Equations for Heat Flow in the Canister.

Since there is no phase change in the canister, and since for this computation the discrete temperatures associated with nodes in the PCM and in the enclosed nickel core are considered to be known, the discrete equations for heat flow in the canister are linear. An SOR scheme with red/black ordering is used to solve this linear system. This forms the inner iteration procedure for the canister module. Of course, the discrete temperatures at nodes in the PCM and in the enclosed nickel core must also be updated at each successive time step, but this is done in separate inner iteration procedures according to a complete, outer iteration strategy. This outer iteration strategy involves the computations for the complete problem including all heat flows, fluid flow and void update calculations and its explanation is beyond the scope of this report.

The linear system to be solved is of the form

$$\left.\begin{aligned} &(1 + CR^{+}_{ijk} + CR^{-}_{ijk} + C\theta^{+}_{ijk} + C\theta^{-}_{ijk} + CZ^{+}_{ijk} + CZ^{-}_{ijk})\, T^{n+1}_{ijk} \\ &- CR^{+}_{ijk}\, T^{n+1}_{(i+1)jk} - CR^{-}_{ijk}\, T^{n+1}_{(i-1)jk} - C\theta^{+}_{ijk}\, T^{n+1}_{i(j+1)k} \\ &- C\theta^{-}_{ijk}\, T^{n+1}_{i(j-1)k} - CZ^{+}_{ijk}\, T^{n+1}_{ij(k+1)} - CZ^{-}_{ijk}\, T^{n+1}_{ij(k-1)} \end{aligned}\right\} = Z^{n}_{ijk}\,, \qquad (19)$$

for ijk running through the indices of the cells in the canister; $i = 0$ and $i = I-1$ with $j = 0, 1, \cdots, J-1$ and $k = 0, 1, \cdots, K-1$, for the inner and outer cylindrical walls respectively; and $k = 0$ and $k = K-1$ with $i = 1, 2, \cdots, I-2$ and $j = 0, 1, \cdots, J-1$, for the end walls respectively. Here the CR's, $C\theta$'s, and CZ's are the coefficients of the corresponding temperatures in equations (6), (7) and (8) respectively multiplied by $\theta\,\Delta t$. The right hand side, Z^{n}_{ijk}, is the sum of T^{n}_{ijk} and $(1-\theta)\Delta t$ times the terms from equations (6), (7) and (8) evaluated at the nth time level, plus, for the cells with $i = I-1$, $(1-\theta)\Delta t$ times the known flux terms at the exterior cylindrical boundary evaluated at the nth time level, plus $\theta\,\Delta t$ times the known flux terms at the exterior

cylindrical boundary evaluated at the $(n+1)$th time level. (Recall that, because of the boundary conditions, some of the coefficients, CR, $C\theta$, and CZ, will be missing or zero, and further that, for cells with $i = 0$, the temperatures of neighbor cells in the nickel core, $i = -1$, are known.)

Since the number of cells in each end disk of the canister and in both the inner and outer cylinders of the canister is odd, we can use the "natural ordering" to number the entire list of cells so that each odd numbered cell has only even numbered cells for neighbors and each even numbered cell has only odd numbered cells for neighbors. With the cells numbered in this way we think of the odd numbered cells as "red" and the even numbered cells as "black." Then any update with red/black ordering consists of an update of the discrete temperatures for red (odd numbered) cells followed by an update for the black (even numbered) cells. The motivation for using the red/black ordering is to make these computations vectorize for fast execution on the Cray X-MP supercomputer.

The SOR iterate for each cell is a linear combination of the Gauss-Seidel update and the previous iterate. The Gauss-Seidel update is computed and then a multiple of the difference between the Gauss-Seidel update and the previous iterate is added to the previous iterate. If we add a new superscript "p" to the discrete temperature variable and use this to denote the iteration number, then the SOR update can be written as

$$T_{ijk}^{n+1,p+1} = T_{ijk}^{n+1,p} + \omega \left(TG_{ijk}^{n+1,p+1} - T_{ijk}^{n+1,p} \right),$$

where TG is the Gauss-Seidel update. The multiplier, ω, is the overrelaxation parameter, a number in $(1, 2)$ for overrelaxation. This parameter is an input variable, so the user may select it as he sees fit. As a result of numerical experiments using the given data of the problem, we settled on a value of 1.3 for ω. The computation is performed for ijk running through the indices of the cells in the canister first for the red nodes using old information (from the previous iteration) at all the black nodes, and then for the black nodes using the latest information at all the red nodes.

The Gauss-Seidel update is computed from the following rearrangement of (19).

$$TG_{ijk}^{n+1,p+1} = \left\{ \begin{array}{l} Z_{ijk}^{n} + CR_{ijk}^{+}\, T_{(i+1)jk}^{n+1,m} + CR_{ijk}^{-}\, T_{(i-1)jk}^{n+1,m} + C\theta_{ijk}^{+}\, T_{i(j+1)k}^{n+1,m} \\ + C\theta_{ijk}^{-}\, T_{i(j-1)k}^{n+1,m} + CZ_{ijk}^{+}\, T_{ij(k+1)}^{n+1,m} + CZ_{ijk}^{-}\, T_{ij(k-1)}^{n+1,m} \,. \end{array} \right. \tag{20}$$

where the superscript m is "p" when red nodes are being updated and "$p+1$" when black nodes are being updated. This iteration is continued until the difference between the Gauss-Seidel update and the previous iterate is less than a specified tolerance.

The computation of Z_{ijk}^{n}, on the right of equation (19), requires evaluating all the difference quotients at the nth time level that would give the result of an explicit update of the enthalpies and temperatures. We take the result of this explicit update, with the maximum stable, explicit, time step size (as computed at the previous time step), as the zeroth iterate for T_{ijk}^{n+1}. The maximum stable, explicit, time step size is time dependent since equivalent thermal conductivities between canister and PCM change as the PCM melts and refreezes. It is determined dynamically while the coefficients that go into the terms of Z_{ijk}^{n} are being computed. However, since this determination is not completed until after at least one iteration of the procedure, the maximum stable time step size lags one time step.

B. Discrete Equations for Energy Redistribution in the PCM.

Since there is a phase change in the PCM, and since the thermophysical parameters of the solid and liquid PCM differ, and since the discrete nonlinear constitutive relation, equation (11), must be satisfied simultaneously with the difference equations, the discrete equations for heat flow in the PCM are nonlinear. A modified SOR scheme with red/black ordering is used to solve this mildly nonlinear system. This forms the inner iteration for the PCM module. For this computation, the discrete temperatures at nodes in the canister and the velocities and void locations are considered to be known.

The nonlinear system to be solved is similar to that represented by equation (19).

$$\left.\begin{array}{l} (1 + C\rho e^{n,p+1}_{ijk,total})\,\rho e^{n,p+1}_{ijk} + CT^{n,p+1}_{ijk,total}\,T^{n,p+1}_{ijk} \\ - C\rho eR^{+n,p+1}_{ijk}\,\rho e^{n,p+1}_{(i+1)jk} - C\rho eR^{-n,p+1}_{ijk}\,\rho e^{n,p+1}_{(i-1)jk} - C\rho e\theta^{+n,p+1}_{ijk}\,\rho e^{n,p+1}_{i(j+1)k} \\ - C\rho e\theta^{-n,p+1}_{ijk}\,\rho e^{n,p+1}_{i(j-1)k} - C\rho eZ^{+n,p+1}_{ijk}\,\rho e^{n,p+1}_{ij(k+1)} - C\rho eZ^{-n,p+1}_{ijk}\,\rho e^{n,p+1}_{ij(k-1)} \\ - CTR^{+n,p+1}_{ijk}\,T^{n,p+1}_{(i+1)jk} - CTR^{-n,p+1}_{ijk}\,T^{n,p+1}_{(i-1)jk} - CT\theta^{+n,p+1}_{ijk}\,T^{n,p+1}_{i(j+1)k} \\ - CT\theta^{-n,p+1}_{ijk}\,T^{n,p+1}_{i(j-1)k} - CTZ^{+n,p+1}_{ijk}\,T^{n,p+1}_{ij(k+1)} - CTZ^{-n,p+1}_{ijk}\,T^{n,p+1}_{ij(k-1)} \end{array}\right| = Z^{n}_{ijk}\,, \quad (21)$$

for ijk running through the indices of the cells in the PCM, $i = 1, 2, \cdots, I-2$, $j = 0, 1, \cdots, J-1$ and $k = 1, 2, \cdots, K-2$. Here the $C\rho eR$'s, $C\rho e\theta$'s and $C\rho eZ$'s are the coefficients of the corresponding ρe's in equations (15), and hence (18), (with the velocities evaluated at the $(n+1)$th time level but with the current iterate from the separate flow update module), multiplied by $\theta\,\Delta t$, and the coefficient $C\rho e^{n,p+1}_{ijk,total}$ is just the sum of these. Similarly the CTR's, $CT\theta$'s and CTZ's are the coefficients of the corresponding temperatures in equations (12), (13) and (14) respectively (with the equivalent thermal conductivities evaluated at the $(n+1)$th time level but with the current iterate for the discrete temperature values, i. e. the pth iterate for the red nodes and the $(p+1)$th iterate for the black nodes,) multiplied by $\theta\,\Delta t$, and the coefficient $CT^{n,p+1}_{ijk,total}$ is just the sum of these. The term on the right hand side of equation (21), Z^{n}_{ijk}, is the sum of ρe^{n}_{ijk} and $(1-\theta)\,\Delta t$ times the difference quotients from equations (12) through (15) evaluated at the nth time level. (Here, because the boundary conditions do not directly affect the heat transfer problem in the PCM, none of the CTR, $CT\theta$, and CTZ coefficients will be missing or zero, but for cells that abut the canister walls, discrete temperatures of the neighboring cells in the canister are known. Furthermore, because of the discrete "no slip" boundary conditions on the velocities at the canister walls, some of the coefficients, $C\rho eR$'s and $C\rho eZ$'s but not $C\rho e\theta$'s, that involve these velocities, will be zero for cells that abut the canister walls.)

Since the number of cells in the inner and outer cylinders of the canister is odd and since the walls of the canister are only one cell thick, the number of cells in each cylinder of PCM is also odd. We apply the same natural numbering to the cells in each cylinder of PCM so that in each cylinder each odd numbered cell has only even numbered cells for neighbors and each even numbered cell has only odd numbered cells for neighbors. Since each cylinder is identically numbered, each cell has identically numbered cells as neighbors in the next innermost and outermost cylinders.

The cylinders of PCM are naturally indexed by the radial index "i." We think of this cylinder index as defining a parity and use this to define the red/black ordering for the complete array of cells in the PCM. In the odd numbered cylinders, we think of the odd numbered cells as "red" and the even numbered cells as "black." While, in the even numbered cylinders, we think of the even numbered cells as "red" and the odd numbered cells as "black." With this assignment each red cell in the entire array has only black cells for neighbors and each black cell in the entire array has only red cells for neighbors. Then, as before, any update with red/black ordering consists of an update of the ρe and temperature variables for the red cells followed by their update for the black cells. Again, the motivation for using the red/black ordering is to make the computations vectorize for the Cray X-MP.

To avoid oscillations about the melt temperature in the iterative solution of the mildly nonlinear implicit equations, Elliot and Ockendon, [1], recommended taking the Gauss-Seidel update as the new iterate for the temperature if the previous iterate and the Gauss-Seidel update are on opposite sides of the melt temperature but using an SOR iteration if the previous iterate and the Gauss-Seidel update are on the same side of the melt temperature. They were analyzing a finite element scheme instead of a finite difference scheme, but the motivation for making this decision is independent of that distinction. We have taken their advice and selectively applied the SOR scheme in the computation of the

discrete temperature in each cell. We defer momentarily discussing this selective application. After the new iterate for the temperature has been determined, the ρe variable at the advanced time step is computed using this temperature in a rearranged and slightly reorganized version of equation (21).

As usual, the SOR iterate for the discrete temperature in each cell in the PCM is a linear combination of the Gauss-Seidel update and the previous iterate. The Gauss-Seidel update is computed by solving equations (11) and (21) simultaneously. A unique solution of these equations exists, and is easily computed, since the temperature is a nondecreasing function of the enthalpy. This computation affects the evaluation of the Gauss-Seidel update, but it does not change the SOR scheme. If the SOR iterate is to be used, then a multiple of the difference between the Gauss-Seidel update and the previous iterate is added to the previous iterate. This multiplier is the overrelaxation parameter, ω. Elliot and Ockendon, [1], give a rather complicated suggestion for determining a good value for the overrelaxation parameter ω. We have not taken their suggestion, but used the same value input for the canister module, 1.3. The SOR update is computed from

$$T_{ijk}^{n+1,p+1} = T_{ijk}^{n+1,p} + \omega \left[TG_{ijk}^{n+1,p+1} - T_{ijk}^{n+1,p} \right],$$

where TG is the Gauss-Seidel update, for ijk running through all the nodes of the PCM, first for the red nodes using old information (from the previous iteration) at all the black nodes, and then for the black nodes using the latest information at all the red nodes.

The Gauss-Seidel update is computed from the simultaneous solution of equations (11) and (21) as follows. We rearrange equation (21) so that only the two terms involving ρe_{ijk}^{n+1} and T_{ijk}^{n+1} are on the left, add and subtract appropriate terms involving T_{melt} and rewrite the resulting equation in a form suitable for iterative solution as

$$(1 + A)\,\rho e_{ijk}^{n+1,p+1} + B\,(T_{ijk}^{n+1,p+1} - T_{melt}) = [\,]_{ijk}^{n}(T_{melt}, 0), \qquad (22)$$

where "A" and "B" denote the negatives of the sums of the coefficients of ρe_{ijk}^{n+1} and T_{ijk}^{n+1}

in the right hand side of equation (10) respectively (as identified in equations (12) - (18)), "p" is the iteration counter, and $[\,]_{ijk}^{n\,m}(T_{melt}, 0)$ denotes the right hand side of equation (10) *with* T_{ijk}^{n+1} *and* ρe_{ijk}^{n+1} *replaced by* T_{melt} *and zero respectively throughout*, and the superscript "m" indicates that the mth (where m is "p" if the ijkth node is a red node and "$p+1$" if it is a black node) iterate value is to be used for ρe^{n+1} and T^{n+1} with subscripts other than "ijk." Although the coefficients A and B do depend on node location and time level, for economy of notation, we have omitted their subscripts and superscripts, (which would include "m" as well as "n"). The computation of the equivalent thermal conductivities is done as discussed after equation (21). Note that both of the coefficients A and B are positive.

To solve equations (11) and (22) simultaneously, we consider the three, mutually exclusive possibilities that the updated state of the ijkth cell is solid, liquid or mush. In the first case: $\rho e_{ijk}^{n+1,p+1} \leqslant 0$, $T_{ijk}^{n+1,p+1} \leqslant T_{melt}$, equation (11) is

$$\rho e_{ijk}^{n+1,p+1} - \rho_{solid}\, c_{solid}\, (T_{ijk}^{n+1,p+1} - T_{melt}) = 0,$$

and the solution is

$$T_{ijk}^{n+1,p+1} = T_{melt} + \frac{[\,]_{ijk}^{n\,m}(T_{melt}, 0)}{(1+A)\,\rho_{solid}\, c_{solid} + B}.$$

In the second case: $\rho_{liquid}\, H \leqslant \rho e_{ijk}^{n+1,p+1}$, $T_{melt} \leqslant T_{ijk}^{n+1,p+1}$, equation (11) is

$$\rho e_{ijk}^{n+1,p+1} - \rho_{liquid}\, c_{liquid}\, (T_{ijk}^{n+1,p+1} - T_{melt}) = \rho_{liquid}\, H,$$

and the solution is

$$T_{ijk}^{n+1,p+1} = T_{melt} + \frac{[\,]_{ijk}^{n\,m}(T_{melt}, 0) - (1+A)H}{(1+A)\,\rho_{liquid}\, c_{liquid} + B}.$$

In the third case, $0 < \rho e_{ijk}^{n+1,p+1} < \rho_{liquid}\, H$, and $T_{ijk}^{n+1,p+1} = T_{melt}$ is the second condition, equation (11), and the solution also.

The quantity $[\,]_{ijk}^{n\,m}(T_{melt}, 0)$ determines the choice from among the three possibilities. If $[\,]_{ijk}^{n\,m}(T_{melt}, 0) \leqslant 0$, then the first alternative is selected and $T_{ijk}^{n+1,p+1} \leqslant T_{melt}$. If

$(1 + A)\rho_{liquid} H \leq [\,]_{ijk}^{n m}(T_{melt}, 0)$, then the second alternative is selected and $T_{melt} \leq T_{ijk}^{n+1,p+1}$. Finally, if $0 < [\,]_{ijk}^{n m}(T_{melt}, 0) < (1 + A)\rho_{liquid} H$, then the third alternative is selected, and $T_{ijk}^{n+1,p+1} = T_{melt}$.

The computational strategy is as follows. First $[\,]_{ijk}^{n m}(T_{melt}, 0)$ is evaluated and then used to compute the corresponding Gauss-Seidel update for the temperature as one of the three alternatives just given. To avoid oscillations about the melt temperature, SOR is selectively applied at each node as follows . If the previous iterate and the Gauss-Seidel update are on the same side of the melt temperature, then SOR is used; but if the previous iterate and the Gauss-Seidel update are not on the same side of the melt temperature or if either of these is exactly the melt temperature, then the Gauss-Seidel update is taken as the new iterate. After the new value for the temperature has been determined, the new value for ρe is computed using equation (22) rearranged in the form

$$\rho e_{ijk}^{n+1,p+1} = \left[[\,]_{ijk}^{n m}(T_{melt}, 0) - B\,(T_{ijk}^{n+1,p+1} - T_{melt}) \right] / (1 + A). \qquad (23)$$

This insures that the updated temperatures and enthalpies are self consistent. This iteration is continued until the difference between the Gauss-Seidel update and the previous iterate is less than a specified tolerance.

As in the iteration for the determination of discrete temperatures in the canister, the results of an explicit update, with the maximum stable step size (as computed at the previous time step), are used as the zeroth iterates for ρe_{ijk}^{n+1} and T_{ijk}^{n+1}. This computation is almost free since the determination of Z_{ijk}^{n}, on the right of equation (21), requires evaluating all the needed difference quotients at the nth time level. Since thermal conductivities of cells in the PCM change as the PCM melts and refreezes, the maximum stable explicit time step size is time dependent. It is determined dynamically while computing the coefficients that go into the terms of Z_{ijk}^{n}. But, since its determination is not completed until one step of the iteration has been completed, it does lag one time step.

The experimentally determined maximum, economically feasible, implicit time step size for the SOR scheme is about twenty times the maximum stable explicit time step size. At this step size, convergence requires about ten to fifteen iterations and the implicit scheme is cost effective. That is, since one iteration of the implicit scheme requires about as much computation as one explicit update would, an implicit scheme that converges in fifteen iterations with a time step size equal to twenty times the maximum explicit time step size requires only about seventy five percent as much computation as an explicit scheme would. The iteration will converge for larger time steps, but the number of iterations increases so that the computation is not economical when compared with an explicit scheme. (This brief analysis does not tell the whole story. When the transition from insolation to earth shadow occurs, a significant transient is introduced and the number of iterations necessary for convergence increases significantly. But in the midst of either of these periods with constant incoming flux, the process is physically much more stable and the number of iterations required for convergence drops dramatically. Thus on the average the implicit scheme is much more economical than an explicit one would be.)

5. VECTORIZATION FOR THE Cray X-MP

In the sense of documenting the algorithms implemented, the following section is complete. As a description of generally applicable techniques for vectorizing finite difference schemes for numerical solution of partial differential equations, it is only an introduction. Based on what was learned from this effort, we have begun timing and optimization studies on strategies for vectorizing such schemes. But that is another story.

Vector processors, such as the Cray family of supercomputers, excel at performing identical sequences of operations on long vector operands. Thus the strategy for vectorization is to arrange computations so that they consist of such sequences. Logical tests within a loop could lead to different paths through the loop and hence to different sequences of computations in the loop. Thus a loop containing such a logical branch will

not vectorize. In addition, only the innermost in a set of nested loops will vectorize. Thus nested loops should be avoided wherever conveniently possible. Finally, there is a certain initialization cost associated with starting a vector computation, so it is pointless to vectorize a computation for only a short vector and economically advantageous to maximize the lengths of vector operands. A concerted effort has been made to vectorize the computations just described for fast execution on the Cray X-MP.

One strategy used to avoid logical branches within a loop is to evaluate a logical variable as a function of the condition to be tested (with values "1" for "true" and "0" for "false") and use it as a multiplier to choose between two possible values (both of which have been computed). For example, the following scrap of code

```
      if ((tgauss.lt.tmelt .and. temp(i).lt.tmelt).or.
     #   (tgauss.gt.tmelt .and. temp(i).gt.tmelt)  ) then
        temp(i) = temp(i) + omega * (tgauss - temp(i))
      else
        temp(i) = tgauss
      endif
```

that would implement successive overrelaxation if the Gauss-Seidel update for a new temperature were on the same side of the melt temperature as the previous iterate but would use the Gauss-Seidel update directly if not, has been replaced by code logically equivalent to

```
tvalue   = temp(i) + omega * (tgauss - temp(i))
fvalue   = tgauss
nt       = one - (tgauss  - tmelt) * (temp(i) - tmelt)
factor   = max(1,nt) - max(0,nt)
temp(i)  = factor * tvalue + (one - factor) * fvalue
```

This code does the same computation with no logical test or branch. The integer "nt" is less than one if the test is satisfied but greater than or equal to one if it is not. The multiplier "factor" is the logical variable used to select the update value for the temperature. "factor" is one if the test is satisfied ($nt < 1$), and zero otherwise ($nt \geq 1$). Both the values are computed, but only one is stored. This example shows the logic of

what has been done. This idea has been used many times in the code with various degrees of subtlety.

An integer Heaviside step function at one is very useful in defining appropriate logical variables. "levi(k) = 1 + max(0,k) - max(1,k)," defines such a function with levi(k) = 1 for $k \geq 1$ and levi(k) = 0 for $k < 1$. In the scrap of code shown above, 1 - levi(nt) was used to define the logical multiplier because the test should be failed if either of the temperatures tested were exactly the melt temperature (which would result in the value of "nt" being one).

Some logical decisions require that choices be made from among more than two alternatives. In such cases logical variables have been combined to select the correct result. For example, to label the cells of PCM as solid (1), mush (2), or liquid (3) the following, nonvectorizable, code could have been used.

```
      do 10 i = 1,lasti
      do 10 k = 1,lastk
      do 10 j = 0,lastj
        if (rhoe(j,k,i).le.zero) then
          lbl(j,k,i) = 1
        else
          if (rhoe(j,k,i).ge.dlheat) then
            lbl(j,k,i) = 3
          else
            lbl(j,k,i) = 2
          endif
        endif
   10 continue
```

Recall from equation (11) that a cell is liquid if ρe is greater than or equal to the product $\rho_{liquid} H$, solid if ρe is less than or equal to zero, and mushy otherwise.

To make this code vectorizable, the if-then-else constructs can be replaced with a single equation involving three Boolean variables using the "levi" function defined above. The following shows how this can be done and in addition how to treat the three dimensional arrays, "rhoe" and "lbl", as if they were one dimensional arrays, thus avoiding the nested loop and simultaneously making longer vector operands.

```
      last = (lastj + 1) * lastk * lasti - 1
      do 10 j = 0,last
        i1      = levi(int(one - rhoe(j,1,1)))
        i3      = levi(int(one + rhoe(j,1,1) - dlheat))
        i2      = (1 - i1) * (1 - i3)
        lbl(j,1,1) = i1 + 2 * i2 + 3 * i3
   10 continue
```

Here i1, i2 and i3 are logical variables for "liquid," "mush," and "solid" respectively, and again the idea of setting the computed quantity equal to a sum of products of logical variables multiplied with the corresponding values has been used. (This loop is offered for demonstration purposes only. Because the computation is trivial, the vectorized version of this loop is slower, for reasonable sized vectors, than the scalar version. This is not true for similar but more computationally intensive loops.)

The reader may object that a smart compiler, or at least a smart operating system, would (or should) detect that the loop index goes beyond its limits. But neither the CFT compile on the Cray X-MP, nor the FORTRAN 77 compiler on our VAX have objected (yet) to compiling or executing such code. Of course it is necessary to arrange loops correctly so that it is the first index of the array that is used to run through the entries. This is necessary because FORTRAN arrays are stored in memory so that the first index varies most rapidly.

Execution of a vectorized loop causes data to be pipelined through special hardware so that multiple sets of the loop instructions (with different index values) are executed simultaneously. Thus, at any given instant while such a loop is being executed, there are multiple values of the running index for which the indicated computations are being done. This may create so called "vector dependencies" that prevent vectorization. The following is an example of a loop that will not vectorize because of these vector dependencies.

```
      do 10 i = 1,imax
        F(i) = C1 * F(i) + C2 * F(i-1) + C3 * F(i+1) + C4 * F(i-idelt) + C5 * F(i+idelt)
   10 continue
```

The compiler will refuse to set up the vector pipelining for such a loop because for a given

value of the running index, "i", the updated values of F(i-1) and F(i-idelt) that would be used to compute F(i) if the loop were executed in scalar mode might be concurrently in the vector pipeline and thus not available.

In the computation of the Gauss-Seidel updates for temperatures in both the canister and the PCM, numerous loops of this form occur. Recall that "i" denotes the radial index, "j" denotes the angular index, and "k" denotes the axial index. The nonvectorizable form of the main loop in the PCM update would be essentially as follows.

```
      do 10 i = 2, imax-1
      do 10 k = 2, kmax-1
      do 10 j = 1, jmax
        test = z(j,k,i) + C1 * tmelt + C2 * ( temp(j-1,k,i) + temp(j+1,k,i))
     #        + C3 * (temp(j,k-1,i) + temp(j,k+1,i)) + C4 * (temp(j,k,i-1) + temp(j,k,i+1))
      if (test.le.0.0) tgauss = tmelt + test / ( C1 + csol )
      if (0.0.lt.test.and.test.lt.dlheat) tgauss = tmelt
      if (dlheat.le.test) tgauss = tmelt + ( test - dlheat ) / ( C1 + cliq )
      if ((tgauss.lt.tmelt .and. temp(j,k,i).lt.tmelt).or.
     #   (tgauss.gt.tmelt .and. temp(j,k,i).gt.tmelt)   ) then
        temp(j,k,i) = temp(j,k,i) + omega * (tgauss - temp(j,k,i))
      else
        temp(j,k,i) = tgauss
      endif
    10 continue
```

The variables "test" and "z" in this loop are $[\ ]_{ijk}^{n\,m}(T_{melt}, 0)$ and Z_{ijk}^{n} from equations (22) and (21) respectively. (Numerous nasty details have been omitted here including ρe terms from equation (21), determination of the temperature dependent values of the coefficients, and the update of ρe after the determination of the temperature. For this explanation, we ignore these.) We have already shown how to avoid logical tests and conditional branches (even with the three possibilities for the value of "test"). Now we show how to circumvent the apparent vector dependencies.

The array in which temperature values are stored, temp(j,k,i), is dimensioned as "temp(0:jmax+1,1:kmax,1:imax)." The angular index was made the first index to make it easier to implement the periodic boundary condition in the angular variable. A copy of T(jmax,k,i) is stored in T(0,k,i) and similarly a copy of T(1,k,i) is stored in T(jmax+1,k,i)

for each pair of indices "i" and "k." We refer to these duplicates as "phantom entries." Their values are updated, i. e. copied over, after each completed iteration.

These phantoms make necessary some additional computations, but they also make possible treating each cylinder of the canister and PCM and each end disk of the canister as a single vector. Whether the time spent on these additional computations is recovered by being able to do a fast vector computation on a longer vector operand is a delicate question that requires individual analysis. For this particular calculation the trade off appeared to be marginally favorable, but cases for which it is not are easily constructed.

To remove the vector dependencies in such loops, "jmax" and "kmax" were required to be odd and the loops were written in a form analogous to the following.

```
      do 10 i = 2, imax - 1
        irb = irb + 1
      if (irb.eq.3) irb = 1
      do 10 j = irb, (jmax + 2)*kmax - 2, 2
        test = C1 * tmelt + C2 * ( temp(j-1,1,i) + temp(j+1,1,i))
     #           + C3 * (temp(j-jmax-2,1,i) + temp(j+jmax+2,1,i))
     #           + C4 * (temp(j,1,i-1) + temp(j,1,i+1))
      ...
     10 continue
```

where for the first pass (when the red entries of the array are updated) "irb" is initially set to zero and for the second pass (when the black entries are updated) "irb" is initially set to one, and the ellipsis indicates the omitted computations rearranged to avoid conditional branches as explained above. The vectorizing FORTRAN compiler still detects the apparent vector dependencies and would refuse to vectorize such a loop except that a compiler directive has been used to force the vectorization. This is legitimate since it is known that the required entries with lower numbered indices are of the opposite color. They cannot be concurrently in the vector pipeline because they are not being updated.

Before and between the two passes through this nested loop, additional computation is done to save the values of certain "real" red entries and to recompute certain phantom red entries. This saving of some entries and recomputing of others is necessary because in the

first pass the red phantoms are updated using values from the wrong neighbor locations. The correct neighbor locations of the red phantoms are also red entries. Thus if their values were not saved before the first execution of the nested loop, then the old values would be unavailable to correctly update the red phantoms. These red phantoms should be correctly updated so that the Gauss-Seidel update of all the black entries will be done with updated information at all red entries. It is not necessary to correctly update the black phantoms because after the black update, in which the phantoms are again updated using values from the wrong neighbor locations, all the phantoms are reloaded with copies of values computed at the corresponding "real" locations

Considerable effort was expended to make the algorithms explained here vectorize. Logical tests and branches were removed, apparent vector dependencies were eliminated or circumvented, and computations were organized to increase the average length of the vector operands. Nevertheless, because they were just too large, most of the loops did not vectorize. There is a limit to the amount of computation that can be done in a vectorized loop, and most of the loops in the canister and PCM modules exceeded this limit. It is clear how to overcome this obstacle. The computations must be broken into smaller pieces with intermediate results stored in long, temporary storage vectors. Lack of time and money prevented our doing this.

6. Acknowledgements.

Special thanks go to Ted Mroz program manager of ASDP NASA, Bob Migra who conceived the development of the NORVEX code, David Namkoong who is the current project monitor, and to Bob Wichner who coordinated the NORVEX development at ORNL. The methods used in this project, in formulating both the continuous and discrete mathematical models of the phase change process and in vectorizing the finite difference schemes, were developed in related research supported by the Applied Mathematical

Sciences Subprogram of the Office of Energy Research, U. S. Department of Energy. Papers documenting these methods are in preparation [2, 4], and support for the research is gratefully acknowledged. The authors also acknowledge informative discussions on various parts of the work with Vasilios Alexiades, John Drake, Charles Romine, and Patrick Worley, all of the Mathematical Sciences Section, Suzanne Lenhart, who is a member of both the Mathematical Sciences Section and the Mathematics Department at the University of Tennessee, and Paul Williams of the Computing and Telecommunications Division.

References

1. Elliott, C. M. and J. R. Ockendon, *Weak and Variational Methods for Moving Boundary Problems*, Pitman, Boston 1982.

2. Flanery, R. E., Wilson, D. G., and M. A. Williams, Vectorization Techniques for Finite Difference Schemes, (in preparation).

3. Williams, M. A. and D. G. Wilson, IMPSOR, A Fully Vectorized FORTRAN Code for Three Dimensional Moving Boundary Problems with Dirichlet or Neumann Boundary Conditions, ORNL 6393, August, 1987.

4. Wilson, D. G., R. E. Flanery, and M. A. Williams, Implicit Difference Schemes for Multidimensional Moving Boundary Problems, (in preparation).

INTERNAL DISTRIBUTION

1.	V. Alexiades	30.	E. G. Ng
2.	B. R. Appleton	31.	G. Ostrouchov
3.	S. A. David	32.	C. H. Romine
4.	J. B. Drake	33.	R. L. Schmoyer
5-9.	R. E. Flanery	34-38.	R. C. Ward
10.	G. A. Geist	39.	R. P. Wichner
11.	L. J. Gray	40.	P. T. Williams
12-13.	R. F. Harbison	41-45.	D. G. Wilson
14.	M. T. Heath	46.	T. Zacharia
15-19.	J. K. Ingersoll	47.	P. H. Worley
20.	W. F. Lawkins	48.	Central Research Library
21.	S. M. Lenhart	49.	K-25 Plant Library
22.	M. R. Leuze	50.	ORNL Patent Office
23-27.	F. C. Maienschein	51.	Y-12 Technical Library
28.	T. J. Mitchell	52.	Laboratory Records - RC
29.	M. D. Morris	53-54.	Laboratory Records Department

EXTERNAL DISTRIBUTION

55. Professor Dr. Güneri Akovali, Chemistry Department, Middle East Technical University, Ankara, TURKEY

56. Professor Alpay Ankara, Chairman Metallurgical Engineering Department, Middle East Technical University, Ankara, TURKEY

57. Dr. Donald M. Austin, ER-7, Applied Mathematical Sciences, Scientific Computing Staff, Office of Energy Research, Office G-437, Germantown, Washington, DC 20545

58. Dr. George I. Bell, T-7 Division, Los Alamos National Laboratory, P. O. Box 1663, Los Alamos, NM 87545

59. Professor Alfredo Bermúdez de Castro, Universidad De Santiago, Dep. de Ecuaciones Funcionales, Facultad de Matematicas, Santiago de Compostela, SPAIN

60. Dr. Alain Bossavit, Electricite de France, Deparmement Traitement de l'Information et Etudes Mathematique, 1 Ave du General de Gaulle, 92141 Clamart, FRANCE

61. Professor John Bruch, Department of Mechanical and Environmental Engineering, University of California at Santa Barbara, Santa Barbara, CA 93106

62. Dr. T. D. Butler, T-3, Hydrodynamics, Los Alamos Scientific Laboratory, P. O. Box 1663, Los Alamos, NM 87545

63. Dr. Bill L. Buzbee, C-3, Applications Support & Research, Los Alamos Scientific Laboratory, P. O. Box 1663, Los Alamos, NM 87545

64. Professor Gunduz Caginalp, Department of Mathematics and Statistics, University of Pittsburgh, Pittsburgh, PA 15260

65. Dr. John R. Cannon, National Science Foundation, 1800 G Street NW, Washington, D. C. 20550

66. Dr. Jagdish Chandra, Director, Mathematics Division, US Army Research Office, P. O. Box 12211, Research Triangle Park, NC 27709

67. Dr. James S. Coleman, Division of Engineering, Mathematical and Geo-Sciences, Office of Basic Energy Sciences, Department of Energy, ER-17, MC G-256, Germantown, Washington, DC 20545

68. Professor David Colton, Mathematical Sciences Department, University of Delaware, Newark, DE 19711

69. Dr. Sam Coriell, U. S. National Bureau of Standards, Mathematical Analysis Section, Washington, DC 20234

70. Dr. James Corones, Ames Laboratory, Iowa State University, Ames, IA 50011

71. Professor John Crank, Thorpe Cottage, 11 Sherwood Avenue, Ruislip, Middlesex HA4 7XL, UNITED KINGDOM

72. Professor Anna Crowley, Department of Mathematics, Royal Military College of Sciences Shrivenham, Swindon Wilts SN6 8LA, UNITED KINGDOM

73. Professor Emanuele DiBenedetto, Mathematics Department, Northwestern University, Evanston, IL 60201

74. Dr. John J. Dorning, Department of Nuclear Engineering and Engineering Physics, Thornton Hall, University of Virginia, Charlottesville, VA 22901

75. Mr. Miles Dustin, NASA Lewis Research Center MS 301-3, 21000 Brookpark Rd., Cleveland, OH 44135

76. Professor Charles M. Elliot, Mathematics Division, Mathematics and Physics Building, Tue University of Sussex, Falmer Brighton BN1 9QH, UNITED KINGDOM

77. Professor Antonio Fasano, Istituto Matematico Ulisse Dini, V. Le Morgagni 67/A, 50134 Firenze, ITALY

78. Professor George Fix, Department of Mathematics, University of Texas, Arlington, TX

79. Professor Michael Fremond, Laboratoire Central des Pontes et Chaussees, Direction Scientific, 58 Boulevard Lefebvre, F-75732 Paris Cedex 15, FRANCE

80. Professor Avner Friedman, Institute for Mathematics and its Applications, University of Minnesota, 56 East River Road, Minneapolis, MN, 55455

81. Professor Robert E. Funderlic, Computer Science Department, North Carolina State University, Raleigh, NC 27695-8206

82. Doçent Dr. Güzin Gökmen, 9 Eylül Üniversitesi, Matematik Bölümü, Bornova KAMPUS, İzmir, TURKEY

83. Dr. Robert M. Haralick, Department of Electrical Engineering, University of Washington, Seattle, WA 98195

84. Professor D. M. Himmelblau, Department of Chemical Engineering, The University of Texas, Austin, TX 78712

85. Professor Dr. Karl Heinz Hoffman, Angewandte Mathematic I, Universität Augsburg, Memminger Strasse 6, D-8900 Augsburg, WEST GERMANY

86. Mr. Steve Johnson, NASA Lewis Research Center MS 500-316, 21000 Brookpark Rd., Cleveland, OH 44135

87. Dr. Hans G. Kaper, Mathematics & Computer Science, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439

88. Dr. Robert J. Kee, Applied Mathematics Division, 8331, Sandia Laboratories, Livermore, CA 94550

89. Dr. W. Kurz, Laboratoire de metallurgie physique, Ecole polytechnique federale de Lausanne, Ch. de Bellerive 34-1007, Lausanne, SWITZERLAND

90. Dr. Andrew Lacey, Department of Mathematics, Heriot Watt University, Edinberg EH144AS, Ecosse, UNITED KINGDOM

91. Professor Peter D. Lax, Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012

92. Dr. Alex Lehoczky, ES72 Space Science Laboratory, Marshall Space Flight Center, Huntsville, AL 35812

93. Professor Carlos G. Levi, Depatment of Mechanical & Environmental Engineering, University of California, Santa Barbara, CA 93106

94. Professor Enrico Magenes, Instituto di Analisi Numerica, Palazzo Universita, Corso Carlo Alberto 5, 27100 Pavia, ITALY

95. Mr. Bob Migra, SVEDRUP Technologies, 16530 Commerce Court, Midpark Branch, Middleburg Heights, OH 44130

96. Mr. Ted Mroz, NASA Lewis Research Center MS 301-3, 21000 Brookpark Rd., Cleveland, OH 44135

97. Mr. David Namkoong, NASA Lewis Research Center MS 301-3, 21000 Brookpark Rd., Cleveland, OH 44135

98. Professor H. N. Narang, Department of Computer Science, Tuskegee University, Hunting Memorial Building Rm 108D, Tuskegee, AL 36088

99. Dr. Basil Nichols, T-7, Mathematical Modeling and Analysis, Los Alamos Scientific Laboratory, P. O. Box 1663, Los Alamos, NM 87545

100. Professor Marek Niezgodka, Systems Research Institute, ul Newelska 6, 01-447 Warszawa, POLAND

101. Professor John R. Ockendon, Oxford University Computing Center, 24 St. Giles, Oxford OX1 3LB, UNITED KINGDOM

102. Professor Dr. Cevdet Öğüt, Dokuz Eylül Üniversitesi, Fen Bilimleri Enstitüsü, MÜDÜRLÜĞÜ, Bornova KAMPUS, İzmir, TURKEY

103. Major C. E. Oliver, Division of Engineering, Mathematical and Geosciences, Office of Basic Energy Sciences, U. S. Department of Energy, ER-15, Room G-350, Germantown Building, Washington, DC 20545

104. Dr. Ronald Peierls, Applied Mathematics Department, Brookhaven National Laboratory, Upton, NY 11973

105. Professor M. Primicerio, Inst. Matematico Ulisse Dini, Via Morgagni 67/A, I-50134 Firenze, ITALY

106. Professor Henning Rasmussen, Department of Applied Mathematics, University of Western Ontario, London, Ontario N6A 5B9, CANADA

107. Dr. Raoul Robert, Laboratoire I. M. A. G., B. P. No 68, 38402 Saint Martin d' Heres Cedex, FRANCE

108. Professor José Francisco Rodrigues, C.M.A.F., Avenida Prof Gama Pinto 2, 1699 Lisboa, PORTUGAL

109. Dean Robert F. Sekerka, Carnegie Mellon University, Schenley Park, Pittsburgh, PA 15213

110. Professor Dr. Ekrem Selçuk, Assistant Dean, Faculty of Engineering, Middle East Technical University, Ankara, TURKEY

111. Professor Subrata Sengupta, Department of Mechanical Engineering, University of Miami at Coral Gables, Coral Gables, FL 33124

112. Professor Bernard Sherman, Department of Mathematics, New Mexico Technical University, Socorro, NM 87801

113. Mr. Ray Skarda, SVEDRUP Technologies, 16530 Commerce Court, Midpark Branch, Middleburg Heights, OH 44130

114. Professor Ralph E. Showalter, Department of Mathematics, University of Texas, Austin, TX 78712

115. Dr. Robert Siegel, Head, Analytical Fluid Mechanics Section, NASA Lewis Research Center, Cleveland, OH 44135

116. Dr. Alan D. Solomon, 90 Eshel Street, Omer 84965, ISRAEL

117. Professor Dr. Jürgen Sprekels, Fachbereich 10 - Bauwesen, Universität-Gesamthochschule Essen, Postfach 103764, Universitätstrasse 2, D-4300 Essen, WEST GERMANY

118. Dr. Frank Szofran, ES72 Space Science Laboratory, Marshall Space Flight Center, Huntsville, AL 35812

119. Dr. Karl Hermann Tacke, Sonnerbergstrasse 7, 8134 Adliswil, SWITZERLAND

120. Professor Alan B. Tayler, Oxford University Computing Center, 19 Parks Road, Oxford OX13 P1, UNITED KINGDOM

121. Professor Domingo Alberto Tarzia, Programme de Mathematical Pura y Applicada, Instituto de Matematica "Beppo Levi," Universidad Nacional de Rosario, Avenida Pelligrini 250, 2000 Rosario, ARGENTINA

122. Dr. Samuel L. Thompson, Computational Physics and Mechanics, Sandia National Laboratory, Albuquerque, NM, 87185

123. Associate Professor Dr. Macit Toksoy, Dokuz Eylül Üniversitesi, FEN BİLİMLERİ ENTİTÜSÜ, MÜDÜRLÜGÜ, Bornova KAMPUS, İzmir, TURKEY

124. Professor Rohit Trivedi, Ames Laboratory, Iowa State University, Ames, IA 50011

125. Dr. John van der Hoek, The University of Adelaide, G. P. O. Box 498, Adelaide, SOUTH AUSTRALIA 5001

126. Professor Vaughan R. Voller Mineral Resources Research Center, University of Minnesota, 56 East River Road, Minneapolis, MN, 55455

127. Professor Burton Wendroff, Mathematics Division, Los Alamos Scientific Laboratory, Los Alamos, NM 97544

128. Professor Mary Fanet Wheeler, Department of Mathematics, Rice University, Houston, TX 77001

129. Professor Robert E. White, Department of Mathematics, North Carolina State University, Raleigh, NC 27650

130. Margaret A. Williams, IBM Corp., Department 41UD, MS276, Kingston, N. Y. 12401

131. Professor Dr. Tolga Yarman, Fizik Bölümü, Anadolu Üniversitesi, Eskişehir, TURKEY

132. Professor Dr. Numan Zengin, Ege Üniversitesi, FEN Fakültesi, Fizik Bölümü, Bornova KAMPUS, İzmir, TURKEY

133. Office of Assistant Manager for Energy Research and Development, U. S. Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37830

134-144. Office of Scientific and Technical Information, P. O. Box 62, Oak Ridge, TN 37830